

Program and Implementation of Stack Data Structure in C++ Using Arrays

Posted by [Lalith Rallabhandi](#) at 11:55

Stack is an **Abstract Data Type** and common Data Structure . Stack is an ordered list which is used in performing the the operations such as Insertion (Push) , Deletion (Pop) , Top can be performed easily.

Stack refers to "**Last In First Out**" Principle (LIFO)

Applications of Stack :

As there are many applications in the field of computer science . Here I will list few major applications of the stack .

1. Stack Data structures is mainly used in Evaluating the Expressions and Syntax parsing .
2. Stack Data Type is used in conversion of Decimal number to Binary number.
3. Stack is used in the application of the Quick Sort to sort a given array or List . Quick sort is one of the efficient sorting techniques which is based on Divide and conquer algorithm .
4. BackTracking is other major applications of stack . In the maze problems backtracking helps to trace the previous path and each path is stored in the form of stack data structure .

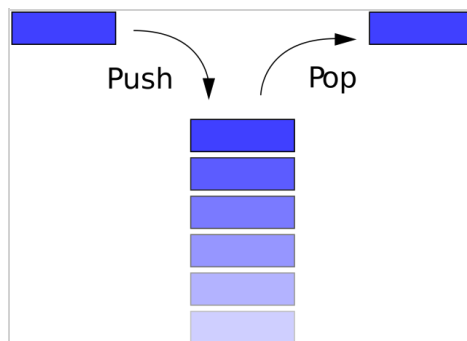


Image credits:my.opera.com

SEARCH THE BLOG

FOLLOW US

STAY CONNECTED

Subscribe **FREE** updates on your Email | [RSS](#)

Like 465

Like us on Facebook

g+

46

Circle us on Google+

Follow [@lalithr95](#)

8,255 followers

Computer Program...



Follow

+1

+ 46

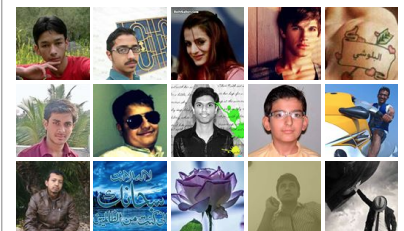
PROGRAMMER



Computer Programming

Like

7,550 people like Computer Programming.



Facebook social plugin

Push :

Push is an operation or function which helps in inserting an element into the stack .

Pop :

Pop is an operation similar to Deletion which helps in deleting or removing an element from the stack .

Top :

Top is a function used to display topmost element in the stack .

Also check : [Implementation and program for complete Binary Tree using C++](#)

Program for Stack and Implementation Using C++

```
#include<iostream>
using namespace std;
template <class T>
class stack
{
public :
    T *a;
    int top,size;
    stack()
    {
        top=-1;
        cout<<"\nEnter size of array : "<<endl;
        cin>>size;
        a=new T[size];
    }
    int isfull()
    {
        if(top==(size-1))
            return 1;
        else
            return 0;
    }
}
```

Recent Activity



[How to create a Password Protected Folder using a Batch Script - Techfinite](#)
3 people recommend this.



[Algorithm and Program for Depth First Search in C - Techfinite](#)
2 people recommend this.



[Complete Binary Tree Implementation and Program in C++ - Techfinite](#)
13 people recommend this.



[Program and Implementation of Stack Data Structure in C++ Using Arrays - Techfinite](#)
35 people recommend this.



[How to Change your Facebook white Blue theme to other themes ? - Techfinite](#)
3 people recommend this.



[Program for Single Linked List Data Structure in C++ and Algorithm - Techfinite](#)
20 people recommend this.



[Program for Queue Data Structure in c++ | Queue Implementation - Techfinite](#)
14 people recommend this.



[Program for Binary Search in C++ and its Algorithm Efficiency - Techfinite](#)
4 people recommend this.



[Program for Bubble Sort in C++ | Programming - Techfinite](#)
8 people recommend this.

Facebook social plugin

```
int isempty()
{
    if(top== -1)
        return 1;
    else
        return 0;
}

void topp()
{
    if(isempty())
        cout<<"\nStack Underflow"<<endl;
    else
        cout<<"\nTop Element is "<<a[top-1]<<endl;
}

void push()
{
    T n;
    if(isfull())
        cout<<"\nStack Overflow"<<endl;
    else
    {
        cout<<"\nEnter an element"<<endl;
        cin>>n;
        a[top++]=n;
        cout<<"\nElement Inserted Succesfully"<<endl;
    }
}

void pop()
{
    if(isempty())
        cout<<"\nStack Underflow"<<endl;
    else
    {
        top=top-1;
        cout<<"\nElement Deleted successfully"<<endl;
    }
};

int main()
{
    stack <int>s;
    int i=0,k;
    while(i!=1)
    {
```

```

cout<<"\n*****M E N U*****\n";
cout<<"1.Push\n2.Pop\n3.Top\n4.Exit\n";
cout<<"\n*****\n";
cout<<"\nEnter option ";
cin>>k;
switch(k)
{
    case 1:
        s.push();
        break;
    case 2:
        s.pop();
        break;
    case 3:
        s.topp();
        break;
    case 4:
        i=1;
        break;
    default :
        cout<<"\n----- Wrong Option -----\n";
        break;
}
}
return 0;
}

```

Output of the Program :

```

Enter size of array :
5
*****M E N U*****
1.Push
2.Pop
3.Top
4.Exit
*****
Enter option 1
Enter an element
3
Element Inserted Succesfully
*****M E N U*****
1.Push
2.Pop
3.Top
4.Exit
*****
Enter option 2
Element Deleted successfully

```

The above program is implemented using Templates and Object Oriented Programming in c++ .

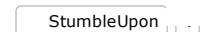
If I have missed anything Please get it to me through comments . Also comment your logic and idea for improvement of the program . If you feel the program for stack can be implemented in another best way please comment below .



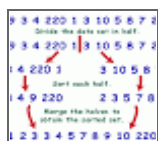
Like {170}



Google +



You Might Also Like:



Merge sort
Algorithm and
Program in...



Quicksort Program
in C & C++ and AI...



Insertion sort
Algorithm and
Insert...



Algorithm and
Program for Depth
Fir...

Labels: [Algorithm](#), [C++](#), [Data Structures](#), [Programming](#)

0 COMMENTS:



POST A COMMENT



Enter your comment...

Comment as:

Google Account

Publish

Preview

[Newer Post](#)

[Home](#)

[Older Post](#)

[Home](#) [About Us](#) [SiteMap](#) [Contact Us](#) [Privacy Policy](#)

[Back to top](#)

DMCA PROTECTION



ALEXA RANK

